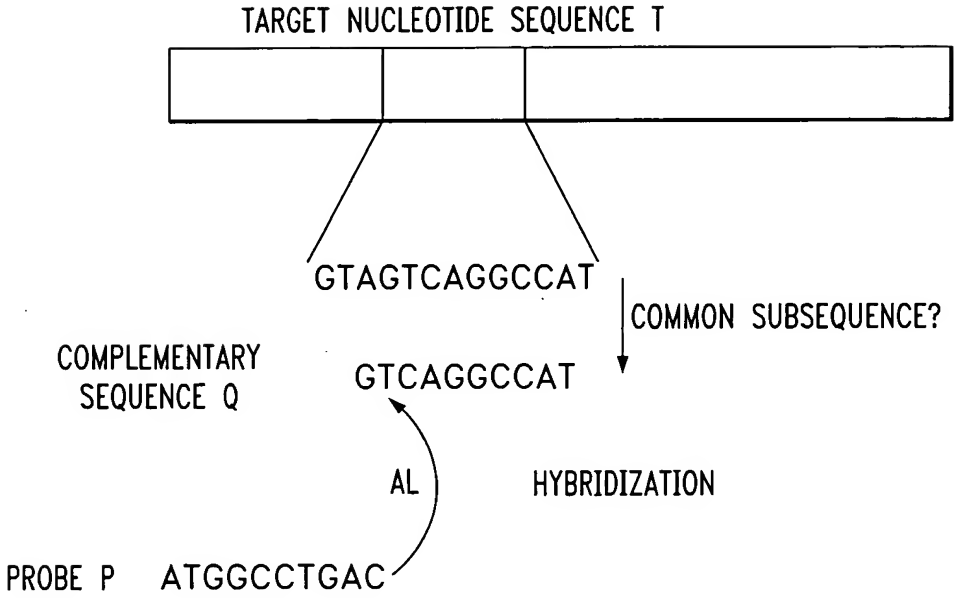
OIPE JC1300
APR 1 2 2004
PATENT & TRADEMARK OFFICE

FIG. 1

(a)



(b)

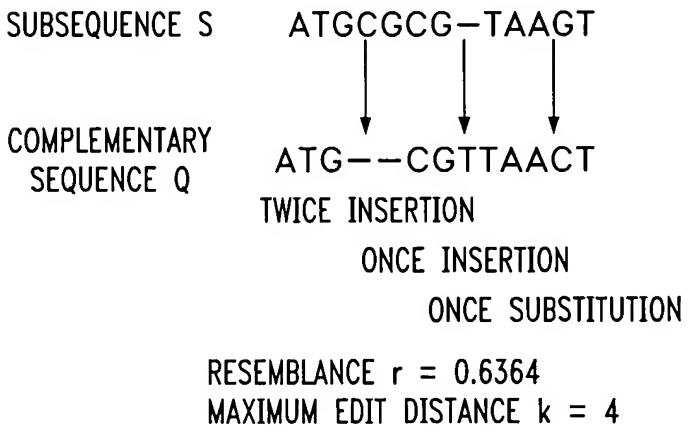


FIG. 2

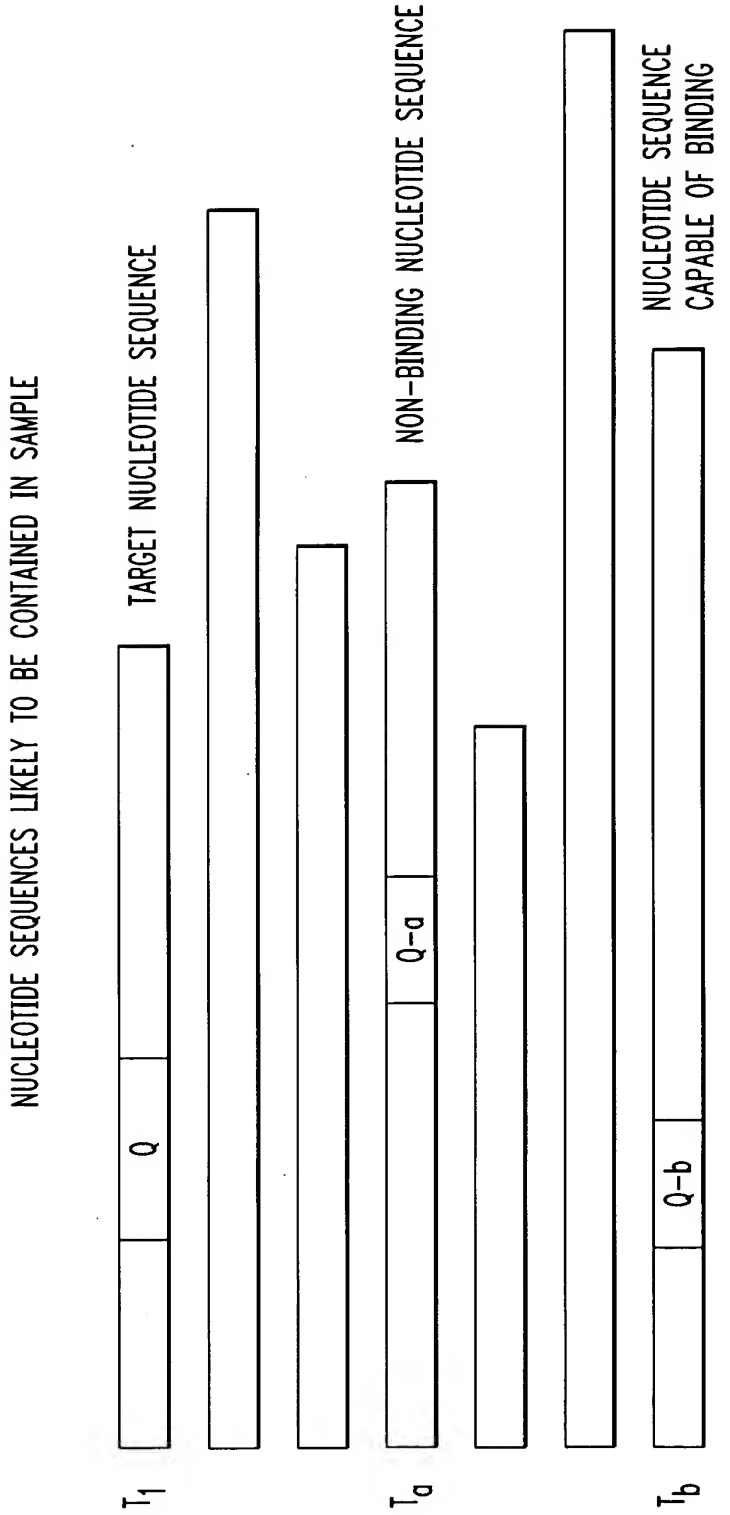

NUCLEOTIDE SEQUENCES LIKELY TO BE CONTAINED IN SAMPLE
TARGET NUCLEOTIDE SEQUENCE
NON-BINDING NUCLEOTIDE SEQUENCE
NUCLEOTIDE SEQUENCE CAPABLE OF BINDING
Q
Q-a
Q-b
T_1
T_a
T_b

FIG. 3

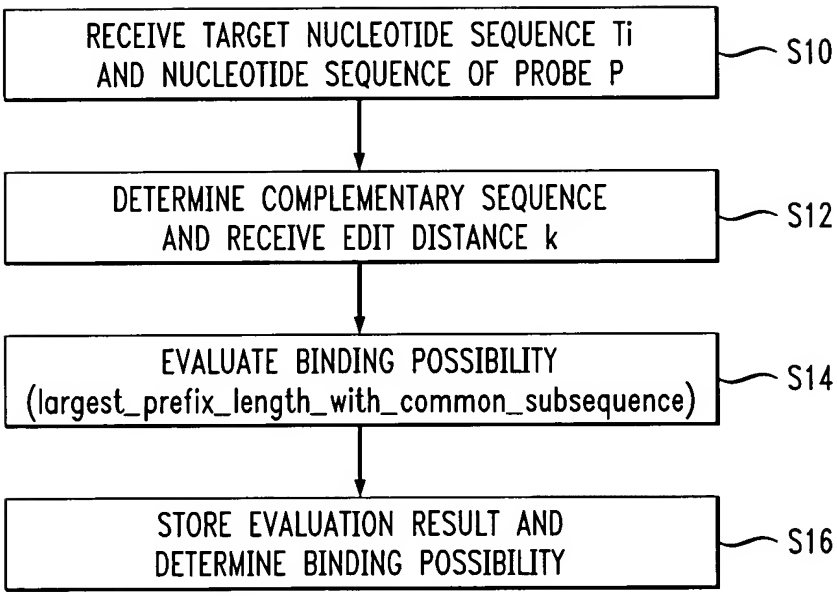

RECEIVE TARGET NUCLEOTIDE SEQUENCE Ti AND NUCLEOTIDE SEQUENCE OF PROBE P
S10
DETERMINE COMPLEMENTARY SEQUENCE AND RECEIVE EDIT DISTANCE k
S12
EVALUATE BINDING POSSIBILITY (largest_prefix_length_with_common_subsequence)
S14
STORE EVALUATION RESULT AND DETERMINE BINDING POSSIBILITY
S16

FIG. 4

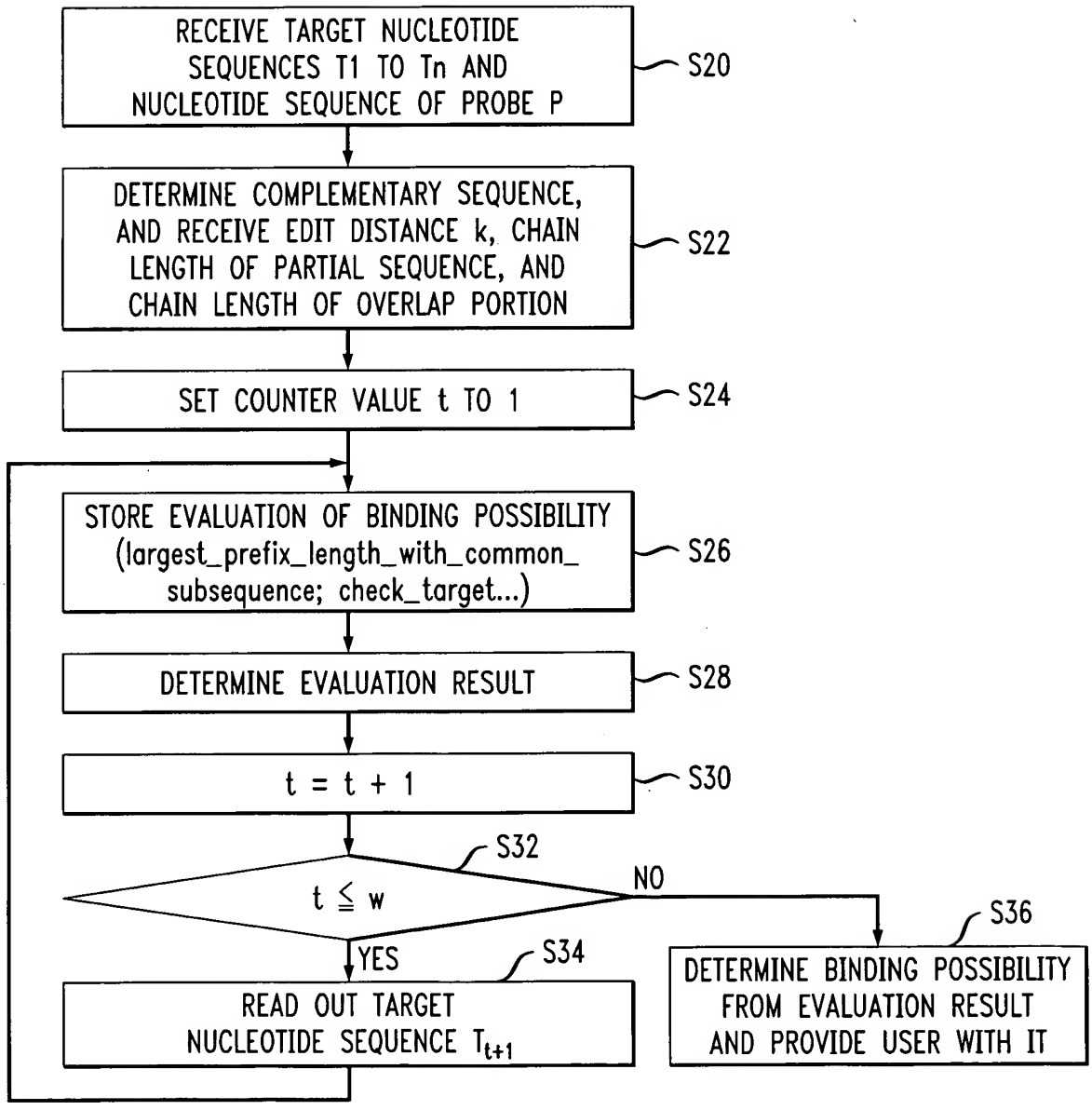

RECEIVE TARGET NUCLEOTIDE SEQUENCES T1 TO Tn AND NUCLEOTIDE SEQUENCE OF PROBE P
S20
DETERMINE COMPLEMENTARY SEQUENCE, AND RECEIVE EDIT DISTANCE k, CHAIN LENGTH OF PARTIAL SEQUENCE, AND CHAIN LENGTH OF OVERLAP PORTION
S22
SET COUNTER VALUE t TO 1
S24
STORE EVALUATION OF BINDING POSSIBILITY (largest_prefix_length_with_common_subsequence; check_target...)
S26
DETERMINE EVALUATION RESULT
S28
t = t + 1
S30
S32
t ≦ w
NO
YES
S34
READ OUT TARGET NUCLEOTIDE SEQUENCE T_{t+1}
S36
DETERMINE BINDING POSSIBILITY FROM EVALUATION RESULT AND PROVIDE USER WITH IT

FIG. 5

function: largest_prefix_with_common_subsequence(T,Q,k)

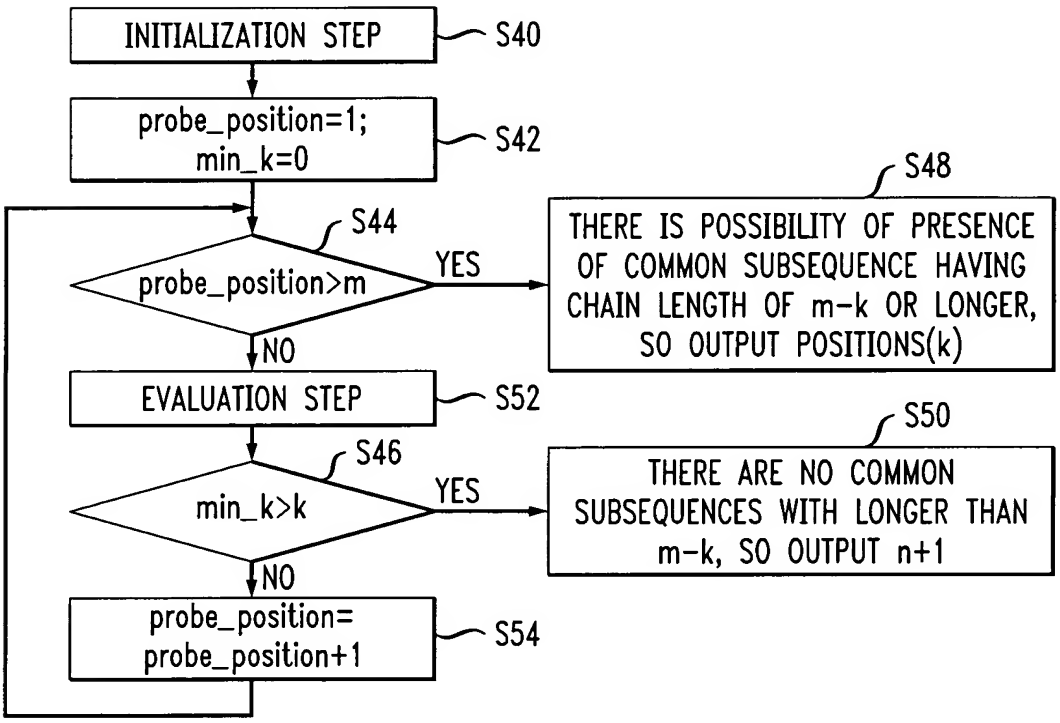


FIG. 6

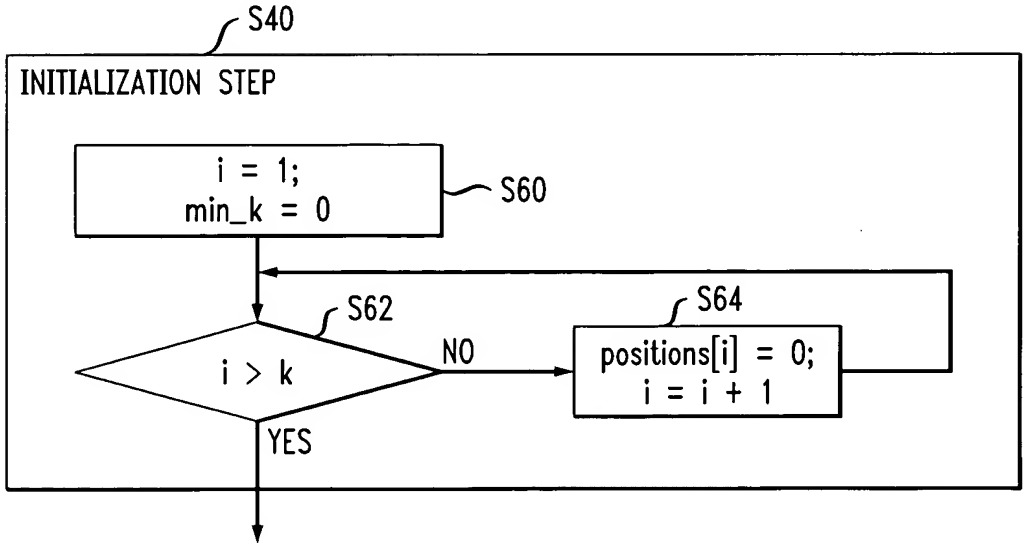


FIG. 7

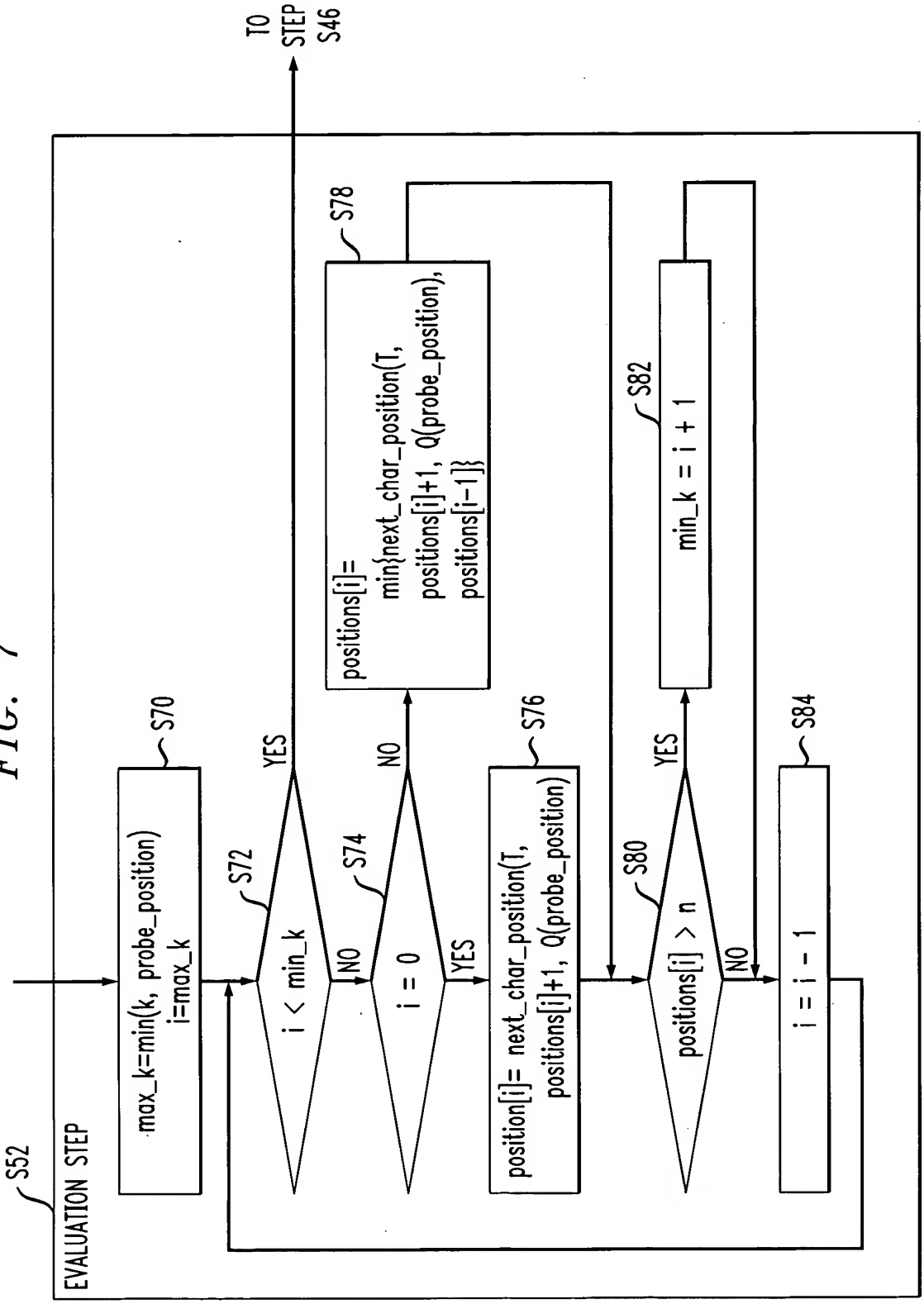

S52
EVALUATION STEP
max_k=min(k, probe_position)
i=max_k
S70
i < min_k
YES
NO
S72
i = 0
NO
YES
S74
positions[i]=
min{next_char_position(T,
positions[i]+1, Q(probe_position),
positions[i-1]}
S78
position[i]= next_char_position(T,
positions[i]+1, Q(probe_position)
S76
positions[i] > n
YES
NO
S80
min_k = i + 1
S82
i = i - 1
S84
TO
STEP
S46

FIG. 8

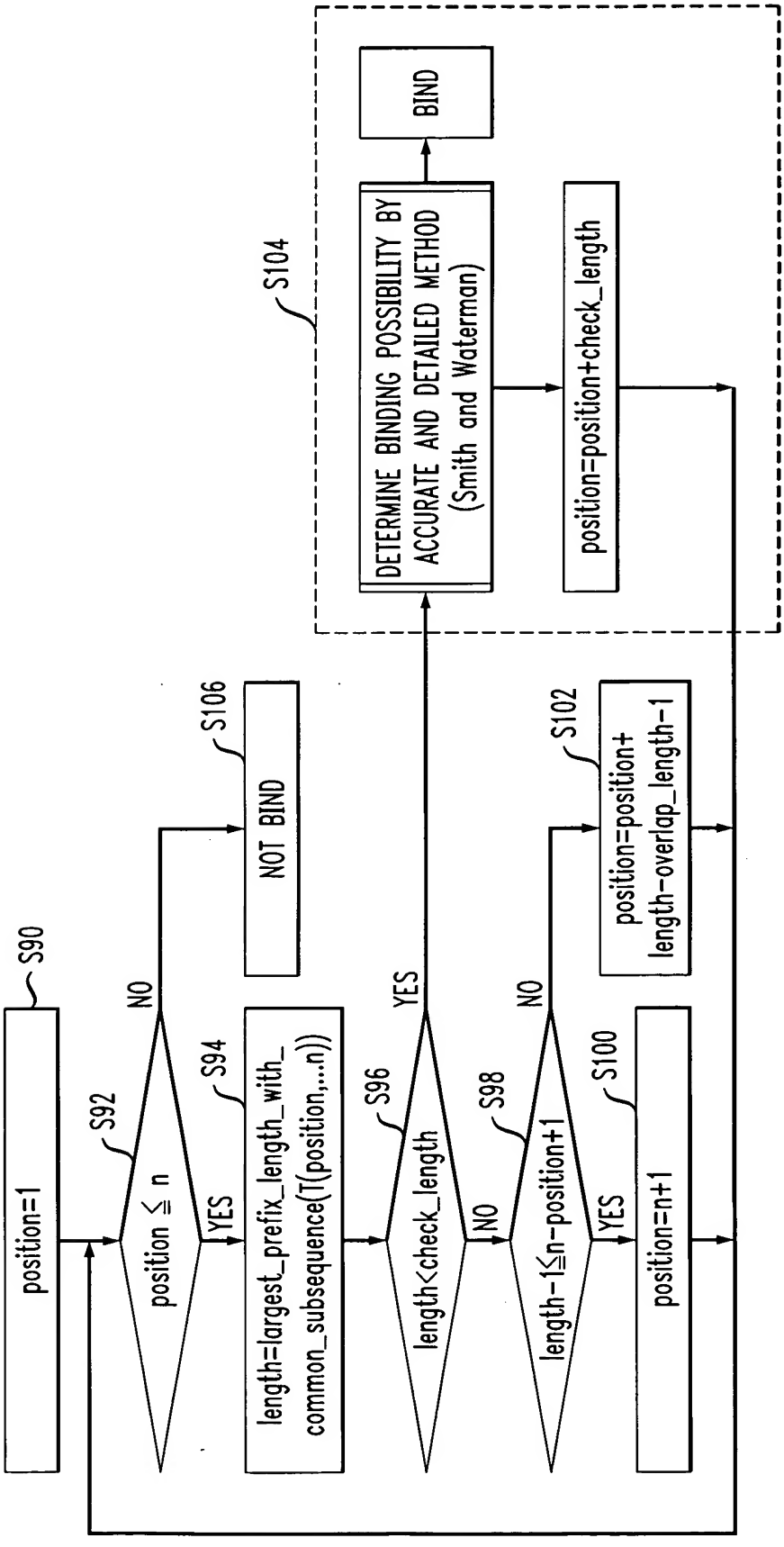

position=1
S90
position ≦ n
S92
NO
YES
NOT BIND
S106
length=largest_prefix_length_with_common_subsequence(T(position,...n))
S94
length<check_length
S96
YES
NO
length-1≦n-position+1
S98
NO
YES
position=position+length-overlap_length-1
S102
position=n+1
S100
S104
DETERMINE BINDING POSSIBILITY BY ACCURATE AND DETAILED METHOD (Smith and Waterman)
BIND
position=position+check_length

FIG. 9

```
//function largest_prefix_with_common_subsequence

largest_prefix_with_common_subsequence(sequence T[1..n], sequence Q[1..m], k) {
    for (i=0;i<=k;i++) positions[i] = 0;
    min_k = 0;
    for (probe_position=1; probe_position<=m; probe_position++) {
        max_k = min {k, probe_position};
        for (i=max_k; i>=min_k ;i--) {
            if (i==0 {
                    positions[i] = next_char_position(T, positions[i]+1,
Q[probe_position]);
            } else {
                    positions[i] = min { next_char_position(T, positions[i]+1,
Q[probe_position]),
                            positions[i-1] }
            }
            if (positions[i] > n) {
                    min_k = i+1;
            }
        }
        if (min_k > k) {
            (1) Absence of common subsequence with length of m-k or longer
            return m+1;
        }
    }
    (2) Presence of common subsequence with length of m-min_k or longer
    return positions(k)
```

FIG. 10

```
//function check_target

check_target(sequence T[1..n], sequence Q[1..m],k, overlap_length, check_length) {
    position=1;
    while(position <= n) {
        length = largest_prefix_length_with_common_subsequence(
                                T[position..n], Q[1..m], k);
        if (length < check_length) {
            if (check_exactly(
                        T[position..(min(n, position+check_length))], Q, k)) {
                position = position + check_length;
            } else {
                return ("T binds to P")
            }
        } else if (length -1 <= n - position + 1){
          position = position + length - overlap_length - 1;
        } else {
          position = n+1;
        }
      }
    return ("T does not bind to P");
}
```

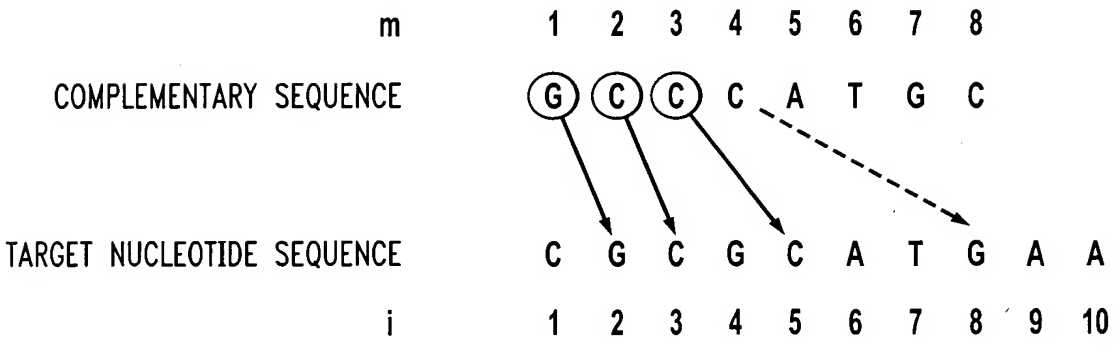
FIG. 11

	1	2	3	4	5	6	7	8
POSITIONS[0]	2	3	5	-	-	-	-	-
POSITIONS[1]	0	1	3	5	6	7	8	-
POSITIONS[2]	-	0	1	3	5	6	7	8
POSITIONS[3]	-	-	0	1	3	5	6	7

FIG. 12

(a)

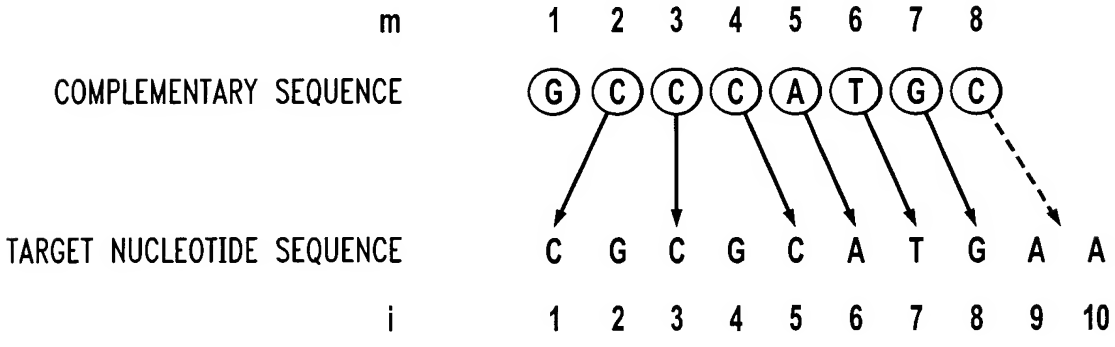
EDIT DISTANCE k=3; positions[0]



[positions[0],m]= (positions[0], 2, 3, 5, –, –, –, –, –)

(b)

EDIT DISTANCE k=3; positions[1]



[positions[1],m]= (positions[1], 1, 3, 5, 6, 7, 8)

FIG. 13

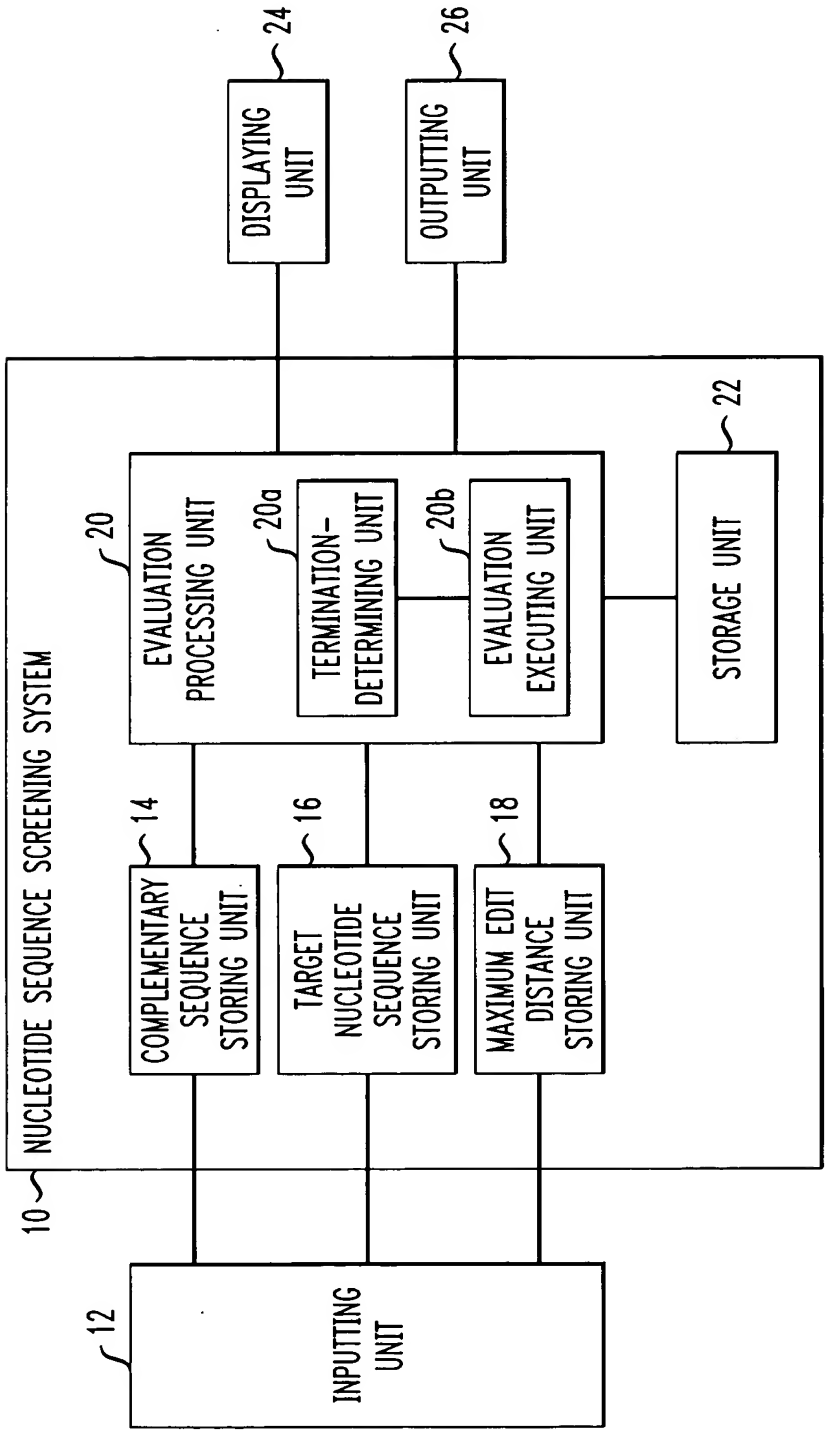
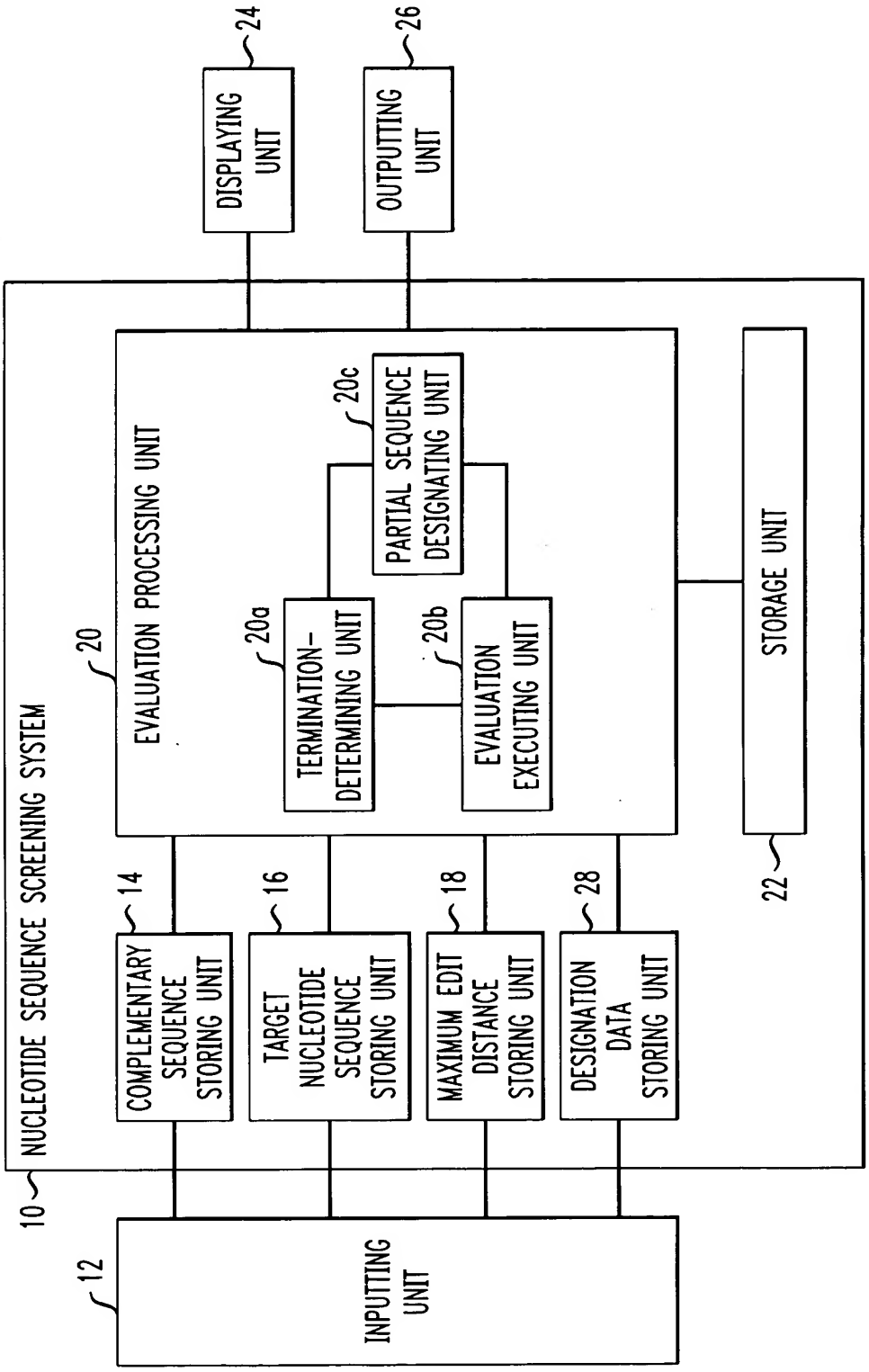

10
NUCLEOTIDE SEQUENCE SCREENING SYSTEM
12
INPUTTING UNIT
14
COMPLEMENTARY SEQUENCE STORING UNIT
16
TARGET NUCLEOTIDE SEQUENCE STORING UNIT
18
MAXIMUM EDIT DISTANCE STORING UNIT
20
EVALUATION PROCESSING UNIT
20a
TERMINATION-DETERMINING UNIT
20b
EVALUATION EXECUTING UNIT
22
STORAGE UNIT
24
DISPLAYING UNIT
26
OUTPUTTING UNIT

FIG. 14


NUCLEOTIDE SEQUENCE SCREENING SYSTEM
10
DISPLAYING UNIT
24
OUTPUTTING UNIT
26
EVALUATION PROCESSING UNIT
20
PARTIAL SEQUENCE DESIGNATING UNIT
20c
TERMINATION-DETERMINING UNIT
20a
EVALUATION EXECUTING UNIT
20b
STORAGE UNIT
22
COMPLEMENTARY SEQUENCE STORING UNIT
14
TARGET NUCLEOTIDE SEQUENCE STORING UNIT
16
MAXIMUM EDIT DISTANCE STORING UNIT
18
DESIGNATION DATA STORING UNIT
28
INPUTTING UNIT
12

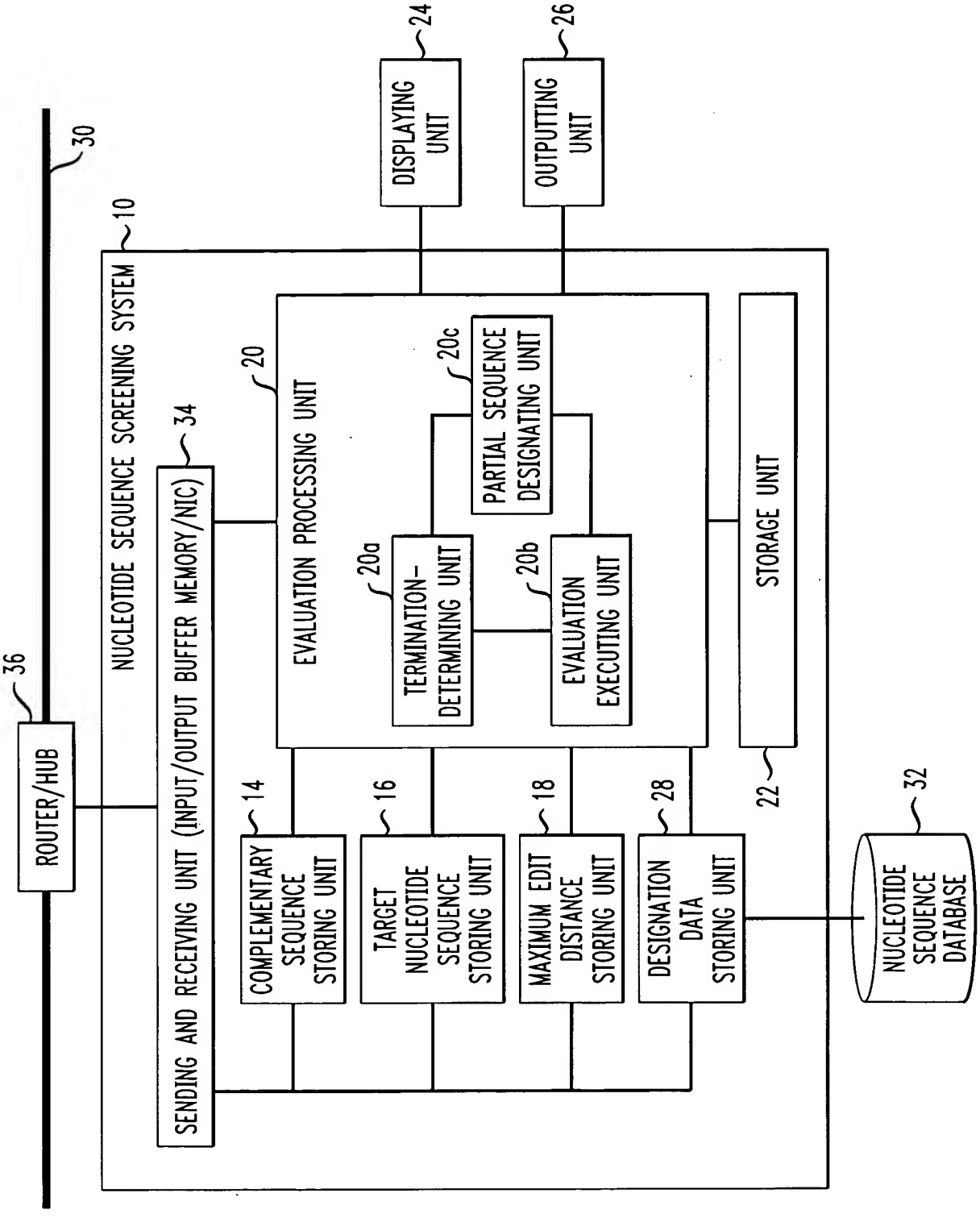

30
10
NUCLEOTIDE SEQUENCE SCREENING SYSTEM
24
DISPLAYING UNIT
26
OUTPUTTING UNIT
36
ROUTER/HUB
34
SENDING AND RECEIVING UNIT (INPUT/OUTPUT BUFFER MEMORY/NIC)
20
EVALUATION PROCESSING UNIT
20c
PARTIAL SEQUENCE DESIGNATING UNIT
20a
TERMINATION-DETERMINING UNIT
20b
EVALUATION EXECUTING UNIT
22
STORAGE UNIT
14
COMPLEMENTARY SEQUENCE STORING UNIT
16
TARGET NUCLEOTIDE SEQUENCE STORING UNIT
18
MAXIMUM EDIT DISTANCE STORING UNIT
28
DESIGNATION DATA STORING UNIT
32
NUCLEOTIDE SEQUENCE DATABASE

FIG. 15

FIG. 16

EXAMPLE				COMPARATIVE EXAMPLE
RESEMBLANCE				Smith-Waterman
0.80	0.85	0.90	0.95	----
405.47s	284.12s	136.20s	45.05s	1108.33s

FIG. 17

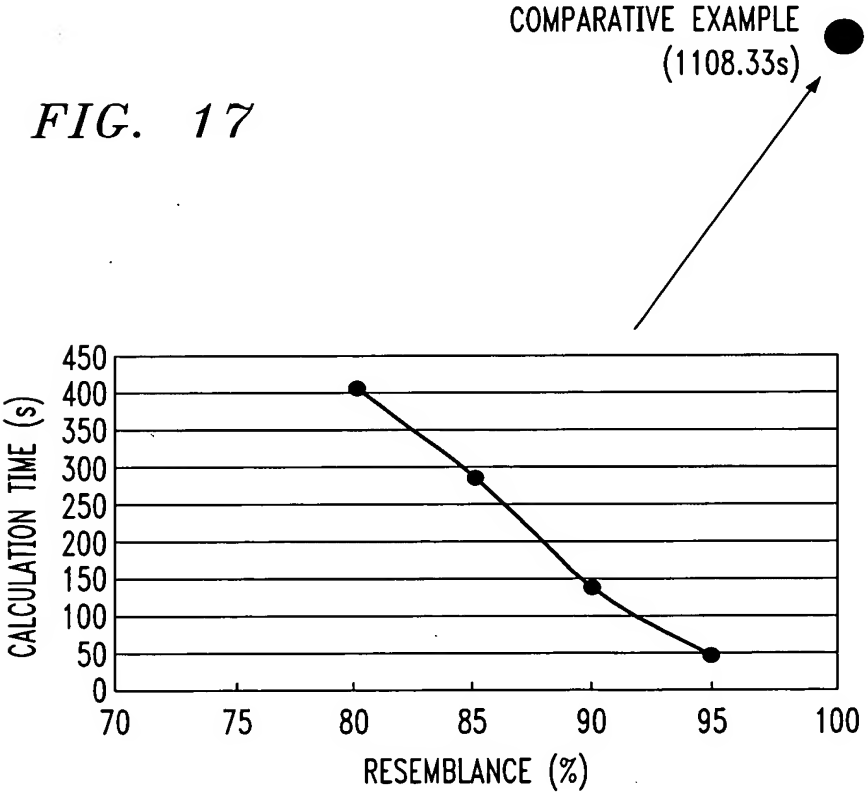
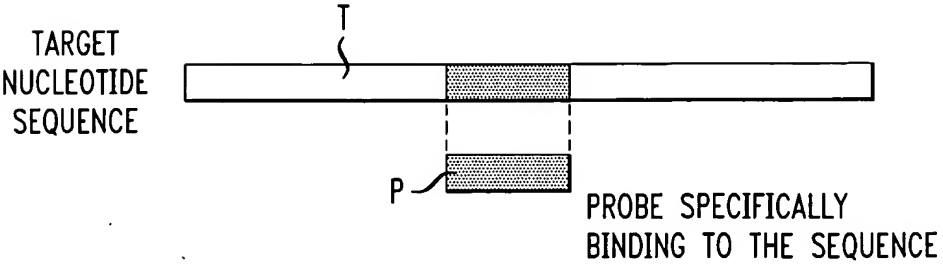



FIG. 18